Refine Search

Search Results -

Terms	Documents
6574761[pn]	2

Database: US Pre-Grant Publication Full-Text Database; US Patents Full-Text Database; US OCR Full-Text Database; EPO Abstracts Database; JPO Abstracts Database; Derwent World Patents Index; IBM Technical Disclosure Bulletins

Search: L12

Refine Search | Recall Text | Clear | Interrupt

Search History

DATE: Thursday, April 14, 2005 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
DB=USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR			
L12	6574761[pn]	2	L12
L11	6256758[pn] or 6550030[pn] or 6474761[pn] or 6530049[pn]	8	L11
L10	L9 and l2	8	L10
L9	L7 same (fault$ or test$)	71	L9
L8	L7 and l2	29	L8
L7	L6 same interconnect$	766	L7
L6	group near3 wire	11530	L6
L5	L1 same interconnect$	1	L5
L4	L1 and l2	0	L4
L3	L1 same wire	2	L3
L2	(field adj1 programmable) or fpga	13694	L2
L1	faulty adj1 group	54	L1

END OF SEARCH HISTORY

US006474761B1

(12) **United States Patent**
Müterthies et al.

(10) **Patent No.: US 6,474,761 B1**
(45) **Date of Patent: Nov. 5, 2002**

(54) **MOUNTING UNIT**

(75) Inventors: **Ralf Müterthies**, Löhne (DE); **Stefan Rüter**, Bad Oeynhausen (DE); **Carsten Meyer**, Bielefeld (DE); **Gerhard Schröder**, Bad Oeynhausen (DE); **Jörg Aufderheide**, Spenge (DE)

(73) Assignee: **Paul Hettich GmbH & Co.**, Kirchlengren (DE)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/663,182**

(22) Filed: **Sep. 15, 2000**

(30) **Foreign Application Priority Data**

Sep. 17, 1999 (DE) 199 44 654

(51) **Int. Cl.**[7] **A47B 88/00**
(52) **U.S. Cl.** **312/348.1**; 312/330.1
(58) **Field of Search** 312/330.1, 334.1, 312/334.7, 334.14, 334.8, 348.1, 348.2, 348.4

(56) **References Cited**

U.S. PATENT DOCUMENTS

3,029,357 A		4/1962	Williams	312/263
3,549,301 A		12/1970	Harris et al.	312/330.1
3,729,246 A		4/1973	Harrell et al.	312/330.1
5,197,791 A		3/1993	Domenig	312/348.1
5,370,454 A		12/1994	Domenig	312/348.1
5,433,518 A		7/1995	Skov	312/348.1
5,906,346 A	*	5/1999	Lin et al.	411/508 X
5,978,232 A	*	11/1999	Jo	312/223.2 X
5,992,958 A	*	11/1999	Scheible	312/330.1 X
6,053,592 A	*	4/2000	Sutterlutti	312/348.1 X

FOREIGN PATENT DOCUMENTS

DE	G 73 17 344.1	5/1973	
DE	GM 79 00 396	6/1980	
DE	G 83 33 251.0	3/1984	
DE	3911353	* 10/1990	312/348.1
DE	40 16 452 A1	9/1991	
DE	G 93 11 493.1	11/1993	
DE	G 94 07 813.0	9/1994	
DE	19726466 A1	12/1998	
EP	551575	* 7/1993	312/348.2

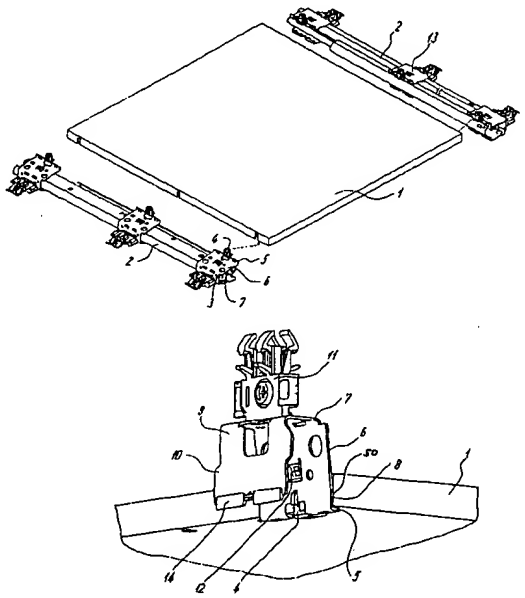
* cited by examiner

Primary Examiner—James O. Hansen
(74) *Attorney, Agent, or Firm*—Barnes & Thornburg

(57) **ABSTRACT**

A mounting unit used for fastening a drawer to a pull-out slide formed of several rails which are displaceably disposed with respect to one another. The mounting unit has several connection elements, each having horizontally extending upper supporting edge which can be connected, at least partially, with an upper edge of an upper rail of the pull-out slide. Each connecting element also includes a downward extending center section and a horizontally extending bottom section on which the sliding bottom is at least partially held. This approximately Z-shaped construction distributes stress well, and reduces costs of manufacture because the connection elements are identical and can be used for the right and left side of a drawer.

13 Claims, 4 Drawing Sheets



L10: Entry 4 of 8 File: USPT Mar 13, 2001

DOCUMENT-IDENTIFIER: US 6202182 B1
**** See image for Certificate of Correction ****
TITLE: Method and apparatus for testing field programmable gate arrays

Abstract Text (1):
A method of built-in self-testing field programmable gate arrays (FPGAs) including the programmable logic blocks, the programmable routing networks and the programmable input/output cells or boundary ports at the device, board or system level includes testing the programmable logic blocks, reconfiguring a first group of he programmable logic blocks to include a test pattern generator and an output response analyzer, and configuring the programmable routing network into groups of wires under test. This step is followed by generating test patterns propagated along the wires under test and comparing the outputs utilizing the output response analyzer. Based on the result of the comparison a pass/fail test result indication is routed to the associated boundary port. The results from a plurality of output response analyzers can be compared utilizing an iterative comparator in order to reduce the number of boundary ports required during testing.

Brief Summary Text (2):
The present invention relates generally to the field of testing of integrated circuit devices and, more particularly, to a method of testing field programmable gate arrays.

Brief Summary Text (4):
A field programmable gate array (FPGA) is a type of integrated circuit consisting of an array of programmable logic blocks (PLBS) interconnected by a programmable routing network and programmable input/output cells. Programming of the logic blocks, the routing network and the input/output cells or boundary ports is selectively completed to make the necessary interconnections that establish one configuration thereof to provide the desired system operation/function for a particular application

Brief Summary Text (5):
The present inventors have recently developed methods of built-in self-testing the array of PLBs in FPGAs at the device, board and system levels. These methods are set out in detail in U.S. Pat. No. 5,998,907 and U.S. Pat. No. 6,003,150. The fill disclosures in these patent applications are incorporated herein by reference.

Brief Summary Text (6):
In each of these prior methods, the reprogrammability of an FPGA is exploited so that the FPGA is configured exclusively with built-in self-test (BIST) logic during testing and subsequently reconfigured to its normal operating configuration. In this way, testability at every level is achieved without overhead. In other words, the BIST logic simply "disappears" when the FPGA is reconfigured for its normal system function. The only cost or additional hardware required for these testing methods is memory for storing the BIST configuration data required for testing and the normal operating configuration required for subsequently reconfiguring the FPGA. This additional memory, however, may be made a part of the test machine environment, e.g., automatic testing equipment, a central processing unit or a maintenance processor, thereby not involving FPGA resources.

Brief Summary Text (7):
In addition to testing the array of PLBs, complete FPGA testing further requires the testing of the programmable routing network. Heretofore, testing of the programmable routing network was accomplished utilizing externally applied test vectors. While the use of test vectors is effective in testing of the programmable routing network, these tests are applicable only for specific device-level manufacturing tests. Accordingly, a need is identified for testing the entire programmable routing network at the device, circuit board and system levels.

Brief Summary Text (9):
An important aspect of the present invention is to provide methods of built-in self-testing FPGAs including the PLBs and the programmable routing network to achieve a complete test at the device, board or system levels. These methods not only test the global routing network which interconnects the array of PLBs but also the local routing network which brings signals into and out of the individual PLBs within the array.

Brief Summary Text (10):
In accordance with the of the present invention, a method is provided for built-in self-testing a programmable routing network of a field programmable gate array (FPGA). The method may be broadly defined as configuring a first group of programmable logic blocks (PLBs) of the FPGA under test to include at least one test pattern generator (TPG) and at least one output response analyzer (ORA), and further configuring a subset of the programmable routing network into at least two groups of wires under test (WUTs). Upon initiation of the built-in self-test (BIST), the at least one TPG generates test patterns which propagate along the at least two groups of WUTs. The outputs of the at least two groups of WUTs are compared utilizing the at least one ORA which in turn generates a test result indication.

Brief Summary Text (11):
More specifically, the FPGA under test is configured in accordance with a BIST configuration retrieved from memory. Preferably, the BIST configured FPGA includes a first group of PLBs configured as at least one TPG for generating test patterns and at least one ORA for receiving and comparing the test patterns, and at least two groups of WUTs along which the test patterns are propagated.

Brief Summary Text (12):
In accordance with an important aspect of the present invention, and in order to achieve a complete BIST of the programmable routing network of the FPGA under test, the WUTs include wire segments interconnected by configurable interconnect points (CIPs) and a second group of PLBs. There are two basic types of CIPs, including cross-point and break-point, and each generally comprises a transmission gate controlled by a configuration memory bit. When incorporated into the WUTs, the second group of PLBs is specifically configured to allow the propagating test patterns to pass there through without alteration. Advantageously, this allows both the global routing network of the programmable routing network including the CIPs and the local routing structures leading to the PLBs to be tested.

Brief Summary Text (13):
In addition, the BIST configured FPGA may also include PLBs from the first group of PLBs configured to align one of the test patterns propagating along one of the at least two groups of WUTs prior to comparison by the ORA. More specifically, these PLBs are configured as swapper cells which map input test patterns to output test patterns. The need for alignment of one of the test patterns arises from the nature of the logic equations implemented in look-up tables (LUTs) of the ORA, the limitations on the inputs to the PLBs implementing the ORA, and the bus rotations in the WUTs.

Brief Summary Text (16):
Additionally, in order to maintain a low number of reconfigurations of the FPGA under test and a short total testing time during the BIST, parallel testing is utilized. More specifically, the test result indications of several ORAs can be combined utilizing an iterative comparator. Alternatively, the test result indication generated by the at least one ORA can be routed directly to a boundary port of the FPGA under test. Advantageously, this provides information regarding the location of the fault in the FPGA under test, as opposed to the single pass/fail test result indication for the entire test.

Drawing Description Text (4):
FIG. 2A is a schematical illustration showing a typical configuration interconnect point of a field programmable gate array;

Drawing Description Text (5):
FIG. 2B is a schematical illustration of a cross-point configuration interconnect point of a typical field programmable gate array;

Drawing Description Text (6):
FIG. 2C is a schematical illustration of a break-point configuration interconnect point of a typical field programmable gate array;

Drawing Description Text (7):
FIG. 3 is a schematical illustration of a typical programmable logic block of a field programmable gate array;

Drawing Description Text (16):
FIG. 11 is a schematic block diagram of an apparatus utilized to test an FPGA.

Detailed Description Text (3):
A typical field programmable gate array (FPGA) generally consists of an array of programmable logic blocks (PLBs) interconnected by a programmable routing network and programmable input/output (I/O) cells or boundary ports. Such structures are, for example, featured in the Lucent ORCA programmable function unit, in the Xilinx XC4000 configurable logic block and in the ALTERA FLEX 8000 logic element. The programmable routing network of the typical FPGA comprises both a global routing network for carrying signals amongst the array of PLBs and the boundary ports, and local routing networks for carrying signals into and out of the PLBs. The typical global and local routing networks associated with a single PLB are shown in FIG. 6 and are discussed in more detail below.

Detailed Description Text (4):
FIG. 1 shows the preferred basic built-in self-test (BIST) architecture 10 utilized in carrying out the method of the present invention to test the programmable routing network of a typical FPGA. In accordance with the first step of the inventive method, a first group of programmable logic blocks of the FPGA under test are configured in accordance with a BIST configuration retrieved from memory located in automatic testing equipment, a central processing unit or a maintenance processing, for example. The BIST configured FPGA includes at least one test pattern generator (TPG) 12 and at least one output response analyzer (ORA) 14. In addition, a subset of the programmable routing network of the FPGA under test is configured to include at least two groups of wires under test (WUTs) 16, 18.

Detailed Description Text (6):
As shown in FIG. 1, the groups of WUTs 16, 18 connect TPG 12 and ORA 14. Upon initiation of the BIST and in accordance with another step of the inventive method, the at least one TPG 12 generates identical test patterns which propagate along the WUTs 16, 18. The test patterns are received by the at least one ORA 14 and are compared to determine whether a fault exists within the WUTs. Based on the outcome of the comparing step of the method, a test result indication is generated by the

at least one ORA 14 and routed to a boundary port of the FPGA. Advantageously, this allows information regarding the location of the fault to be obtained.

Detailed Description Text (15):
In accordance with the method of the present invention, a complete BIST of the array of PLBs in the FPGA will be run before the BIST of the programmable interconnect network. Since the PLB BIST is also using a large portion of the programmable interconnect, we examined the coverage provided by this testing in order to avoid duplicate testing. Rather than attempt to target the individual routing faults left undetected by the PLB test, it was decided that completely testing the programmable routing network is simpler and more efficient, even if some duplication occurs. As indicated above, the methods for testing the array of PLBs are set out in detail in pending U.S. Pat. No. 5,991,907 and U.S. Pat. No. 6,003,150.

Detailed Description Text (18):
Alternately, the various ORA outputs can be routed directly to the boundary ports to retrieve the results as indicated above. This provides more information regarding the location of the fault in the FPGA as opposed to a single pass/fail test indication for the entire test resulting from the use of an iterative comparator.

Detailed Description Text (19):
As indicated above, FIG. 6 illustrates a simplified view of the routing busses associated with a single PLB designated numeral 92 in an ORCA 2C series FPGA. Horizontal and vertical busses are denoted by h and v, respectively. The suffixes x1, x4, xH, and xL indicate wire segments that extend across 1 PLB, 4 PLBs, half the PLB array, and the full length of the PLB array, respectively, before encountering a break-point CIP or a boundary point of the FPGA (not shown). Direct busses provide connections between adjacent PLBs. The four direct busses are designated dn, ds, de, and dw denoting direct north, south, east, and west, respectively. For every PLB there are two sets of vertical x1 busses and two sets of horizontal x1 busses, designated vx1w, vx1e, hx1n, and hx1s. Several CIPs are available to establish different connections among the wire segments as shown by circle and diamond-shaped symbols. The diamond-shaped symbol 94 of a break-point CIP on a 4-bit bus represents a group of 4 individual break-point CIPs. Similarly and as shown in FIG. 6A, a circle-shaped symbol 96 denoting a cross-point CIP at the intersection of a vertical 4-bit bus with an horizontal 4-bit bus represents a group of 4 individual cross-point CIPs between corresponding wires in the two busses. The square-shaped symbol 98, on the other hand, at the intersection of a 5-bit direct bus with a 4-bit x1 bus represents a more flexible matrix of cross-point CIPs shown in FIG. 6B.

Detailed Description Text (20):
The fault model utilized to test the programmable interconnect network of a typical FPGA in accordance with the method of the present invention, includes CIPs stuck-closed (stuck-on) and stuck-open (stuck-off), wire segments stuck at 0 or 1, open wire segments, and shorted wire segments. Detecting the CIP faults also detects stuck-at faults in the configuration memory bits that control the CIPs as shown generally in FIG. 2A. For generality, both wired-AND and wired-OR faults are considered as possible behavior for shorted wire segments. A stuck-closed CIP creates a short between its two wires.

Detailed Description Text (21):
Since detailed layout information regarding the adjacency relationships between wire segments is typically not available, only rough physical data available in FPGA data books, for example, is utilized to determine bunches of wire segments for testing. A bunch of wire segments is a group of wires that may have pair-wise shorts; but not every wire segment is necessarily adjacent to every other wire segment in the bunch. For example, all the vertical wire segments located between

two adjacent PLB columns (partially shown in FIG. 6) may be treated as a bunch even if not all shorts are physically feasible. Advantageously, this makes the BIST method of the present invention layout-independent and allows the bus rotations which make the adjacency relations among the wire segments of the same bunch change to be ignored during testing.

Detailed Description Text (24):
To perform the method of the present invention, an apparatus comprising a field programmable gate array 99, a support platform 101, a controller 103 and memory 105 may be utilized. In the preferred embodiment shown in FIG. 11, the FPGA is an in-circuit reprogrammable FPGA, such as an SRAM-based FPGA, which stores the configuration bits in SRAM and may be reconfigured an arbitrarily large number of times. In particular, the FPGA is configured to include at least one TPG, at least two groups of WUTs for propagating a test pattern, and at least one ORA for comparing the test patterns. By reconfiguring the FPGA, the programmable routing network can be exhaustively tested without dedicating any of circuitry for built-in self-testing. The support platform may include any necessary means (e.g. a circuit board or test bench) for testing the FPGA at the wafer, device, circuit board, or system levels. In addition, the test controller may be automated test equipment generally known in the art for testing wafers/devices, or a CPU or maintenance processor for board/system level testing. The memory should be sufficient to store the BIST configuration for the FPGA and the functional/operational configuration of the system in order to reconfigure the FPGA after BIST.

Detailed Description Text (27):
The present BIST-based diagnostic approach was utilized in the testing and diagnosis of the programmable interconnect network of an ORCA FPGA. The test consisted of four test sessions which included nineteen phases, summarized in Table 1 in terms of the target faults for each test session.

Detailed Description Text (33):
As shown in FIGS. 9A-9E, test session 3 tests most of the remaining cross-point CIPs for stuck-off faults and the break-point CIPs on the direct busses for stuck-on faults. These BIST configurations utilize the same basic structure shown in FIG. 7A but various busses are interconnected utilizing cross-point CIPs as the test patterns pass from the top of the FPGA to the bottom in a zig-zag pattern instead of straight down a single bus.

Detailed Description Text (35):
During the third and fourth configurations shown in FIGS. 9C and 9D, the BIST structure is inverted with the TPGs at the bottom of the FPGA and the ORAs and swapper cells at the top of the FPGA. These additional configurations are required to insure complete testing of the cross-point CIPs in the PLB which connect the PLB outputs to any of the four direct busses. While one set of WUTs goes through the PLB to connect the two sets of direct busses, the other set of WUTs goes through the bi-directional buffers 116 in the PLB such that these buffers are tested in both directions. Advantageously, as a result of the coverage provided by these configurations, separate rotation and re-application of the configurations in the horizontal direction are not required.

Detailed Description Text (38):
These four test sessions provide complete testing of the programmable interconnect network including the wire segments and their interconnecting CIPs with a total of 15 configurations. This, of course, is comparable with the number of configurations required for a complete PLB test set out in detail in pending U.S. patent applications Ser. Nos. 08/729,117 and 08/595,729. It is important to note that the number of phases does not depend on the size of the PLB array in the FPGA.

Other Reference Publication (3):
W.K. Huang Et Al., "An Approach To Testing Programmable/Configurable Field

Programmable Gate Arrays," Proc. IEEE VLSI Test Symp pp. 450-455, Apr. 28-May 5, 1996.

Other Reference Publication (5):
F. Lombardi Et Al., "Diagnosing Programmable Interconnect Systems For FPGAS," Proc ACM/SIGDA Internationl Symp On FPGA, PP 100-106, 1996.

Other Reference Publication (6):
C. Stroud Et Al., "Evalution Of FPGA Resources For Built In Self Test Of Programmable Blocks," Proc ACM/SIGDA Inter. Symp. On FPGAS PP 107-113, 1996.

Other Reference Publication (7):
C. Stoud Et Al., "Built In Self Test For Programmable Logis Blocks In FPGA,"Proc IEEE VLSI Test Symp, PP 387-392, Apr. 28-May 1, 1996.

Other Reference Publication (8):
C. Stroud Et Al., "Using ILA Testing For Bist In FPGAS," Proc IEEE International Test Conf., PP 68-75, Oct. 20-25, 1996.

Other Reference Publication (9):
C. Stroud Et Al., "Bist Based Diagnostics For FPGA Logic Blocks," Proc. IEEE International Test Conf., PP 539-547, Nov. 1-6, 1997.

CLAIMS:

1. A method of testing a programmable routing network of a field programmable gate array comprising the steps of:

configuring a first group of programmable logic blocks to include at least one test pattern generator and at least one output response analyzer, and a subset of said programmable routing network to include at least two groups of wires under test;

initiating a built-in self-test;

generating test patterns with said at least one test pattern generator;

comparing outputs of said at least two groups of wires under test with said at least one output response analyzer; and

generating a test result indication.

2. The method set forth in claim 1, wherein the step of configuring the subset of said programmable routing network further includes utilizing a group of wire segments and a group of configurable interconnect points to form said at least two groups of wires under test.

3. The method set forth in claim 2, wherein said test patterns are exhaustive and are sufficient to detect all possible faults in said group of wire segments and said group of configurable interconnect points.

4. The method set forth in claim 3, wherein said faults include opens and shorts in said group of wires and stuck-open and stuck-closed configurable interconnect points.

10. The method set forth in claim 1, wherein the step of generating a test result indication includes routing said test result indication of said output response analyzer to a boundary port of said field programmable gate array, whereby information regarding the location of any fault in said field programmable gate array may be obtained.

11. The method set forth in claim 1, further comprising the step of iteratively comparing a first test result indication of a first of said output response analyzers to a second test result indication of a second of said output response analyzers, whereby parallel testing of said at least two groups of wires under test is accomplished independent of the available number of boundary ports of said field programmable gate array and whereby the number of reconfigurations of said field programmable gate array and the total testing time are minimized.

12. A method of testing a field programmable gate array including programmable logic blocks and a programmable routing network comprising the steps of:

testing said programmable logic blocks;

configuring a first group of programmable logic blocks to include at least one test pattern generator and at least one output response analyzer, and a subset of said programmable routing network to include at least two groups of wires under test;

initiating a built-in self-test;

generating a first set of test patterns with said at least one test pattern generator,

comparing outputs of said at least two groups of wires under test with said at least one output response analyzer; and

generating a first test result indication,

whereby said field programmable gate array is completely tested.

18. An apparatus for testing a field programmable gate array including programmable logic blocks and a programmable routing network comprising:

a platform for supporting said field programmable gate array during testing;

a controller in communication with said platform for (a) loading one of a plurality of test configurations in said field programmable gate array, said configuration including at least one test pattern generator, at least two groups of wires under test for propagating test patterns, and at least one output response analyzer for comparing said test patterns propagated along said at least two groups of wires under test, (b) for initiating the built-in self-test and (c) for receiving the subsequent results of the built-in self-test; and

a memory in communication with said controller for storing the plurality of test configurations and an operational field programmable gate array configuration.

19. A field programmable gate array including programmable logic blocks and a programmable routing configured to comprise:

at least one test pattern generator;

at least two groups of wires under test for propagating test patterns;

at least one output response analyzers for comparing said test patterns propagated along said at least two groups of wires under test; and

whereby said programmable routing network at said field programmable gate array is tested.

20. The field programmable gate array configuration set forth in claim 19, further including memory for storing the configurations to provide exhaustive testing of

each subset of said programmable routing network.

21. The field programmable gate array configuration set forth in claim 19, having no dedicated circuitry for built-in self-testing applications.

Previous Doc Next Doc Go to Doc#

L10: Entry 2 of 8 File: USPT Oct 7, 2003

DOCUMENT-IDENTIFIER: US 6631487 B1
TITLE: On-line testing of field programmable gate array resources

Abstract Text (1):
A method of testing field programmable gate array (FPGA) resources and identifying faulty FPGA resources during normal on-line operation includes configuring an FPGA into a working area and an initial self-testing area. The working area maintains normal operation of the FPGA throughout testing and identifying of the resources. Within the initial and subsequent self-testing areas, the FPGA resources are initially tested for faults. Upon detection of a fault in the FPGA resources, the initial self-testing area resources are reconfigured or subdivided and further tested in order to identify the faulty resource. Dependent upon the further test results, the FPGA resources may be further subdivided and tested until the faulty resource is identified. Once the faulty resource is identified, the FPGA is reconfigured to replace unusable faulty resources or to avoid faulty modes of operation of partially faulty resources diagnosed during further testing. In this manner, partially faulty resources are allowed to continue operation in a diminished capacity to enhance fault tolerance. After testing each of the FPGA resources located within the initial self-testing area for faults, identifying the faulty resources, and in some instances diagnosing faulty modes of operation of the faulty resource, the FPGA is reconfigured such that a portion of the working area becomes a subsequent self-testing area and the initial self-testing area replaces that portion of the working area. In other words, the self-testing area roves around the FPGA testing its resources in a continuous manner.

Brief Summary Text (2):
The present invention relates generally to the field of testing of integrated circuit devices and, more particularly, to a method of testing field programmable gate array resources and identifying faulty field programmable gate array resources.

Brief Summary Text (4):
A field programmable gate array (FPGA) is a type of integrated circuit consisting of an array of programmable logic blocks interconnected by a programmable routing network and programmable input/output cells. Programming of the logic blocks, the routing network and the input/output cells is selectively completed to make the necessary interconnections that establish one configuration thereof to provide the desired system operation/function for a particular application.

Brief Summary Text (5):
The present inventors have recently developed methods of built-in self-testing the array of programmable logic blocks and the programmable routing resources in FPGAs at the device, board and system levels. These methods are set out in detail in U.S. Pat. Nos. 5,991,907, 6,003,150, and 6,108,806 and pending U.S. application Ser. No. 09/109,123. The full disclosures in these patents and patent applications are incorporated herein by reference.

Brief Summary Text (6):
In addition to these off-line testing methods, the present inventors have also recently developed methods of on-line testing and fault tolerant operation of

FPGAs. These methods are set out in detail in pending U.S. application Ser. Nos. 09/261,776, 09/405,958, 09/406,219, and 09/611,449. The full disclosure of these patent applications is also incorporated herein by reference.

Brief Summary Text (7):
On-line testing and fault tolerant operation of FPGAs is most important in high-reliability and high-availability applications, such as, space missions, telecommunication network routers, or remote equipment in which adaptive computing systems often rely on reconfigurable hardware to adapt system operation. In such applications, the FPGA hardware must work continuously and simply cannot be taken off-line for testing, maintenance, or repair.

Brief Summary Text (8):
When a fault is detected in the resources of the FPGA hardware of these systems during testing, the faulty resource must be quickly identified and the remaining FPGA resources reconfigured to replace the faulty resource, or a faulty mode of operation of the identified resource diagnosed in order to allow continued operation of the faulty resource in a diminished capacity. Therefore, the steps of testing and identifying FPGA resources, and diagnosing faulty modes of operation in the faulty identified resources must be performed concurrently with normal system operation.

Brief Summary Text (10):
In accordance with the present invention, resources of a field programmable gate array (FPGA) determined to be faulty during testing may now be quickly identified and the remaining FPGA resources reconfigured to replace the faulty resource, or a faulty mode of operation of the identified resource diagnosed in order to allow its continued operation in a diminished capacity, during normal on-line operation of the FPGA. Specifically, when a fault is detected in a group of FPGA resources under test, the FPGA resources are reconfigured such that the resources within the group of resources under test are subdivided and grouped with additional FPGA resources known to be fault free for further testing. Dependent upon the result of the further testing, the steps of reconfiguring and further testing are repeated until the detected faulty resource is identified. Once the faulty resource is identified, the remaining FPGA resources are reconfigured to replace the faulty resource in order to provide fault tolerant operation of the FPGA.

Brief Summary Text (11):
On-line testing of the FPGA resources is accomplished by configuring the FPGA into a working area and an initial self-testing area. The working area maintains normal operation of the FPGA throughout testing. Within the initial and subsequent self-testing areas, however, the FPGA resources are tested for faults. It is initially presumed that all of the resources of the FPGA are fault-free as determined through manufacturing testing.

Brief Summary Text (12):
Within the initial self-testing area, test patterns are generated and applied to FPGA resources selected for testing. Outputs of a first group of FPGA resources under test, e.g., programmable logic blocks, programmable routing network resources, or a combination of both, are preferably compared to outputs of a second group of equivalently configured FPGA resources under test. Based on a comparison of the outputs of the groups of FPGA resources under test, fault status data is generated.

Brief Summary Text (13):
When the fault status data indicates the detection of a fault in the groups of FPGA resources under test, the resources within the initial self-testing area of the FPGA are reconfigured into subsequent groups of FPGA resources under test for further testing in order to identify the faulty resource. As indicated above, the group of FPGA resources under test where the fault was detected is subdivided and

its resources grouped with additional FPGA resources known to be fault free to form subsequent groups of FPGA resources for further testing in the manner described above.

Brief Summary Text (14):
In accordance with another aspect of the present inventive method, the initial self-testing area of the FPGA is preferably configured to include groups of FPGA resources, or self-testing tiles, for testing the programmable logic blocks. Each self-testing tile preferably includes programmable logic blocks configured to function as a test pattern generator, an output response analyzer, and equivalently configured programmable logic blocks under test. The initial self-testing area of the FPGA may be divided into any number of equivalently configured self-testing tiles so long as each testing tile contains a sufficient amount of FPGA resources to complete testing. Advantageously, this allows for concurrent testing of the programmable logic blocks in several testing tiles within the self-testing area, thus reducing the overall test time and fault latency.

Brief Summary Text (16):
When the fault status data indicates the detection of a fault in one of the logic blocks of a testing tile, the resources within the initial self-testing area of the FPGA are reconfigured into subsequent self-testing tiles in order to identify the faulty programmable logic block. Specifically, the logic blocks in the testing tile where the fault is originally detected are subdivided and grouped with additional logic blocks known to be fault-free to form the subsequent testing tiles wherein the logic blocks are further tested in the manner described above. Dependent upon the subsequent fault status data, the programmable logic blocks in the subsequent self-testing tiles may be further subdivided, grouped, and tested until the detected faulty logic block is identified. Once the faulty logic block is identified, the remaining FPGA resources are reconfigured to replace the faulty logic block in order to provide fault tolerant operation. Alternatively, diagnostic testing of the identified logic block is conducted in order to determine a faulty mode of operation.

Brief Summary Text (18):
To test the programmable routing resources of the FPGA, test patterns are generated and propagated along groups of wires under test. Output patterns of a first group of wires under test are preferably compared to output patterns of a second group of wires under test receiving the same patterns. In order to achieve a complete test of the programmable routing resources within the initial self-testing area, the groups of wires under test include wire segments of varying lengths interconnected by configuration interconnect points, and possibly programmable logic blocks configured as identify functions. Advantageously, this allows both global routing resources between programmable logic blocks and local routing resources leading to each programmable logic block to be tested. As indicated above, the output patterns of the groups of wires under test are compared and test result data generated.

Brief Summary Text (19):
In order to minimize the number of reconfigurations of the FPGA under test and reduce overall testing time and fault latency, parallel testing of the programmable routing resources are preferably utilized. Specifically, comparisons of the output patterns of segments of the groups of wires under test may be made at several locations along the groups of wires under test utilizing a plurality of FPGA resources configured as output response analyzers producing fault status data for each segment of the groups of wires under test.

Brief Summary Text (20):
When the fault status data indicates the detection of a fault in one of the groups of wires under test, the resources within the groups of wires under test are reconfigured into subsequent groups of wires under test in order to identify the faulty routing resource. Specifically, the resources in the groups of wires under

test where the fault is originally detected are subdivided and grouped with additional resources known to be fault-free to form the subsequent groups of wires under test wherein the routing resources are further tested in the manner described above. Dependent upon the subsequent fault status data, the routing resources in the subsequent groups of wires under test may be further subdivided, grouped, and tested until the detected faulty routing resource is identified. Once the faulty routing resource is identified, the remaining FPGA resources are reconfigured to replace the faulty resource in order to provide fault tolerant operation.

Brief Summary Text (21):
Upon completion of testing each of the FPGA resources located within the initial self-testing area, identifying the faulty resources, and diagnosing the faulty logic blocks, the FPGA under test is reconfigured so that a portion of the working area becomes a subsequent self-testing area, and the initial self-testing area, reconfigured to avoid faulty resources, becomes a portion of the working area. In other words, the self-testing area roves around the FPGA under test repeating the steps of testing and reconfiguring within the self-testing areas until each portion of the FPGA under test, is reconfigured as a subsequent self-testing area, tested, and detected faults within the FPGA resources diagnosed. As noted above, the present method of testing allows for normal operation of the FPGA under test to continue within the working area uninterrupted by the testing conducted within the self-testing areas.

Brief Summary Text (22):
An apparatus for testing the resources of the FPGA under test during normal on-line operation includes a test and reconfiguration controller in communication with the FPGA under test for: (a) configuring the FPGA under test into an initial self-testing area and a working area, the working area maintaining normal operation of the FPGA; (b) testing the resources located within the initial self-testing area for faults; (c) reconfiguring the resources located within the initial self-testing area for further testing in order to identify the faulty resource; (d) further testing resources located within the reconfigured self-testing area; and (e) repeating the steps of reconfiguring and further testing until the faulty resource is identified. The testing apparatus further includes a storage medium in communication with the test and reconfiguration controller for storing a plurality of test configurations, and usage and fault status data for each FPGA resource.

Brief Summary Text (23):
An FPGA in accordance with the present invention includes a plurality of programmable logic blocks and a plurality of programmable routing resources interconnecting the programmable logic blocks initially configured as an initial self-testing area for testing at least a portion of the programmable routing resources and/or programmable logic blocks for faults, and an initial working area for maintaining normal operation of the FPGA during testing. The portion of the programmable routing resources and/or programmable logic blocks located within the initial self-testing area are further subdivided and tested until the faulty programmable routing resource or programmable logic block is identified.

Drawing Description Text (3):
FIG. 1 is a schematic block diagram of an apparatus for testing resources of a field programmable gate array (FPGA);

Drawing Description Text (4):
FIG. 2 is an illustration of an FPGA under test configured into an initial self-testing area and a working area wherein the working area maintains normal operation of the FPGA under test;

Drawing Description Text (5):
FIG. 3 is an illustration of the FPGA under test configured such that the working area is divided into four disjoint areas by a vertical self-testing area and a

horizontal self-testing area;

Drawing Description Text (6):
FIG. 4 is a schematic block diagram showing a preferred comparison-based self-testing area configured to include a test pattern generator, an output response analyzer, and two groups of FPGA resources under test;

Drawing Description Text (7):
FIG. 5 is an illustration of the preferred FPGA under test with the initial self-testing area divided into several distinct self-testing tiles;

Drawing Description Text (9):
FIG. 7A is an illustration of an initial vertical self-testing tile within the initial self-testing area of the FPGA under test in accordance with a preferred embodiment;

Detailed Description Text (3):
A typical field programmable gate array (FPGA) generally consists of a plurality of resources including an array of programmable logic blocks (PLBs) interconnected by programmable routing resources and programmable input/output cells or boundary scan ports (most FPGAs feature a boundary-scan mechanism). Such structures are, for example, featured in the Lucent ORCA programmable function units, in the Xilinx XC4000 configurable logic block, and in the ALTERA FLEX 8000 logic element. In accordance with the present inventive method, the resources of an FPGA under test 10 are tested for faults, the faulty resources identified, and the FPGA 10 reconfigured to avoid the identified faulty resources during normal on-line operation.

Detailed Description Text (4):
As shown in schematic block diagram in FIG. 1, configuring, testing, identifying, and reconfiguring of the FPGA resources are preferably controlled by a test and reconfiguration controller 12. In the present preferred embodiment, an external test and reconfiguration controller 12 is utilized because present commercially available FPGAs do not allow internal access to their configuration memory. Accordingly, a configuration decompiler tool of a type known in the art is utilized to determine the intended function or mode of operation of the FPGA resources. Alternatively, this information may be extracted from the design stage and made available to the controller 12. It should be appreciated by those skilled in the art that any controller, e.g., internal or external to an FPGA, could be utilized with an FPGA that allows for internal access to its configuration memory and that a single test and reconfiguration controller is capable of controlling several FPGAs. For purposes of illustration of the present preferred embodiment of the invention, however, a one-to-one controller to FPGA ratio is utilized.

Detailed Description Text (5):
The preferred controller 12 may be implemented on an embedded microprocessor in communication with a storage medium or memory 14 for storing the various FPGA operational configurations, as well as, test data, and fault-tolerant functions including their associated fault-tolerant reconfigurations. The controller 12 and memory 14 further exchange and store usage status data (e.g., functional status, unused spare status or spare status) used in testing and fault status data (e.g., fault-free status, defective status or partially usable status) for the FPGA resources. These data are subsequently utilized in reconfiguring the FPGA under test 10 to replace unusable faulty resources or to avoid faulty modes of operation of partially faulty PLBs. The utilization of usage and fault status data is further described in more detail below.

Detailed Description Text (6):
In operation, the controller 12 accesses the FPGA under test 10 in a known manner such that access is transparent to normal function of the FPGA 10. As best shown in

FIG. 2, the FPGA under test 10 is initially configured by the controller 12 into an initial self-testing area 16 and a working area 18. Advantageously, this approach allows for normal operation of the FPGA under test 10 to be maintained within the working area 18 while the FPGA resources are each tested for faults in the initial self-testing area 16.

Detailed Description Text (7):
More specifically, the initial self-testing area 16 is preferably configured to include a vertical self-testing area 20 and a horizontal self-testing area 22 (shown in FIG. 3) which intermittently rove around the FPGA under test 10 during testing. While only one self-testing area is required to test PLBs, both self-testing areas 20 and 22 are required for identifying faulty PLBs or programmable routing resources, and for testing the programmable routing resources.

Detailed Description Text (8):
Depending upon the location of the two self-testing areas 20 and 22 at any given time during testing, the working area 18 may be contiguous, or it may be divided into two or four disjoint regions 24 as shown in FIG. 3. To accommodate testing in the self-testing areas 20 and 22, vertical wire segments in the vertical self-testing area 20 and horizontal wire segments in the horizontal self-testing area 22 are all designated reserved or unusable during operation of the FPGA under test 10. In this manner, connections between PLBs in the disjoint regions 24 of the working area 18 may be made utilizing the horizontal wire segments through the vertical self-testing area 20 and vertical wire segments through the horizontal self-testing area 22.

Detailed Description Text (9):
Testing of the FPGA 10 is generally accomplished by configuring its resources within the initial self-testing area 16 to function as a test pattern generator (TPG) 26 and an output response analyzer (ORA) 28, and as equivalently configured FPGA resources under test 30 as shown in FIG. 4. During testing, equivalent test patterns generated using the TPG 26 are applied to the FPGA resources under test 30. Outputs of the FPGA resources under test 30 are compared by the ORA 28 to determine whether a fault exists within either of the resources under test 30. A match/mismatch result of the comparison performed by the ORA 28 is communicated as a pass/fail result or fault status data through boundary-scan ports of the FPGA 10 (not shown) to the controller 12. The fault status data is stored in memory 14 and utilized by the controller 12 in reconfiguring the FPGA resources for further testing and fault tolerant operation as described in more detail below.

Detailed Description Text (10):
When testing the PLBs as shown in FIG. 5, the initial self-testing area 16 is preferably divided into any number of equivalently configured self-testing tiles 32 so long as each testing tile 32 contains a sufficient amount of FPGA resources to complete testing. In other words, each testing tile 32 must include at least a group of PLBs configured to function as the TPG 26, a group of PLBs configured to function as the ORA 28, and PLBs under test 30. As shown in FIGS. 6A-6C, for example, a typical TPG 26 may include a group of three PLBs (T), and single PLBs (O and B) may be configured to function as the ORA 28 and the PLBs under test 30. By systematically rotating the functions of the PLBs in each self-testing tile 32, as sequentially shown in FIGS. 6A-6F, each PLB becomes a PLB under test 30 exactly twice during the preferred method of testing, each time being compared to a different PLB under test. This type of repeating strategy allows for the detection of both single faulty PLBs, as well as, several concurrently faulty PLBs. In addition, the utilization of several self-testing tiles 32 allows for concurrent testing of PLBs within the tiles 32, thus reducing the overall test time and fault latency.

Detailed Description Text (11):
Complete testing of the PLBs within the self-testing tiles 32 is achieved by

repeatedly reconfiguring the PLBs under test 30 for testing in every possible mode of operation and applying exhaustive test patterns in each mode of operation. Without having a detailed knowledge of the implementation of the FPGA under test 10, the modes of operation of its PLBs may be determined only from the information provided in an associated FPGA data manual. In addition to repeatedly reconfiguring the PLBs under test 30, the TPG 26 is reconfigured by the controller 12 for each new mode of operation of the PLBs under test 30 which require test patterns different from the ones generated for the previous mode of operation. The ORA 28 is also reconfigured by the controller 12 when the new mode of operation of the PLBs under test 30 involves a different number of outputs. It is important to note that all of these reconfigurations occur concurrently with the normal operation of the FPGA under test 10, which is unaffected by and unaware of the testing activity. The present preferred comparison-based on-line method of testing is described in greater detail for programmable logic blocks in pending U.S. application Ser. No. 09/405,958 incorporated herein by reference.

Detailed Description Text (12):
When the fault status data indicates the detection of a fault in one of the PLBs 30 of a testing tile 32, roving of the self-testing areas is initially interrupted in order to identify the faulty PLB and to identify its faulty modes of operation. Specifically, the initial self-testing area 20 or 22 wherein the faulty PLB lies is reconfigured for further testing in order to identify the faulty PLB. Preferably, the FPGA resources contained within an initial self-testing tile are subdivided for inclusion with known fault-free PLBs in two subsequent testing tiles. The PLBs in each subsequent testing tile are independently tested in the manner described above. Dependent upon the subsequent test results, the FPGA resources within one or both of the subsequent testing tiles may be further reconfigured or subdivided and tested until the detected faulty PLB is identified.

Detailed Description Text (15):
As sequentially shown in FIGS. 7A-7D, upon detection of a fault in one of the PLBs (A-F) in an initial self-testing tile 32 configured within an initial vertical self-testing area 33, the FPGA resources within the initial self-testing tile 32 are subdivided for inclusion in two subsequent testing tiles 34 and 36 within the vertical self-testing area 33. As shown in FIG. 7B, PLBs A and B are grouped with additional known fault-free FPGA resources (FF) to form an upper vertical self-testing tile 34, and PLBs C, D, E, and F are grouped with additional known fault-free FPGA resources (FF) to form a lower vertical self-testing tile 36.

Detailed Description Text (16):
For purposes of illustration, assume that the PLBs A and B within the upper vertical testing tile 34 pass all subsequent exhaustive testing but a fault is detected in the PLBs within the lower vertical testing tile 36. The lower vertical testing tile 36 is again subdivided into additional distinct vertical testing tiles 38, 40. As shown in FIG. 7C, PLBs C and D are grouped with additional known fault-free FPGA resources (FF) to form a second upper vertical testing tile 38, and PLBs E and F are grouped with additional known fault-free FPGA resources (FF) to form a second lower vertical testing tile 40. Again, assume that the PLBs within the upper vertical testing 38 pass all subsequent exhaustive testing and the fault is detected in the lower vertical testing tile 40.

Detailed Description Text (17):
As shown in FIG. 7D, the remaining PLBs C and D which potentially contain the detected fault reside in the same row thus precluding further subdividing into distinct vertical self-testing tiles. However, by reconfiguring the self-testing tile 40 into distinct left side and right side horizontal self-testing tiles 42 and 44 utilizing a horizontal self-testing area 45, the remaining PLBs C and D may be tested by grouping them with additional known fault-free FPGA resources (FF) for further testing. At this point, the detected fault is completely identified to PLB C or D dependent upon the results of testing. The test and reconfiguration

controller 12 will use this data to reconfigure the remaining FPGA resources to avoid the identified faulty PLB altogether. Alternatively, diagnostic testing of the identified PLB is conducted in order to determine a faulty mode of operation of the PLBs.

Detailed Description Text (19):
In a manner similar to that described with respect to the testing of programmable logic blocks, the programmable routing resources including both global routing resources for carrying signals amongst the array of PLBs and local routing resources for carrying signals into and out of the PLBs are comparatively tested. Specifically, as shown in FIG. 8, PLBs within the initial self-testing area 16 are configured to function as a TPG 48 and an ORA 50, and a portion of the programmable routing resources are configured to include at least two groups of wires under test (WUTs 52). The groups of WUTs 52 may include wire segments 54, configurable or configuration interconnect points (CIPs 56), and PLBs 58.

Detailed Description Text (20):
During testing exhaustive test patterns generated using the TPG 48 are applied to and propagated along the groups of WUTs 52. The outputs of the groups of WUTs 52 are compared by the ORA 50 to determine whether a fault exists within either group of WUTs 52. A match/mismatch result of the comparison performed by the ORA 50 is communicated as a pass/fail test result or fault status data through the boundary-scan ports of the FPGA under test 10 to the controller 12. The fault status data is stored in memory 14 and utilized by the controller 12 in reconfiguring the FPGA resources for the testing and fault tolerant reconfiguration as described in more detail below. In order to minimize the number of reconfigurations required during testing, and therefore the total testing time, parallel testing of the programmable routing resources is preferably utilized. Specifically, as shown in FIG. 9, comparisons of the output patterns of segments 54 of the groups of WUTs 52 may be made at several locations along the groups of WUTs 52 utilizing more than one ORA 50. Advantageously, this allows one set of TPG generated test patterns to be used to test several differing groups of WUTs per configuration. The present preferred comparison-based on-line method of testing is described in greatest detail for programmable routing resources in pending U.S. application Ser. No. 09/406,219 incorporated herein by reference.

Detailed Description Text (21):
When the fault status data indicates the detection of a fault in one of the groups of WUTs 52 in a self-testing area, roving of the self-testing areas is initially interrupted in order to identify the faulty programmable routing resource. Specifically, the groups of WUTs 52 wherein the faulty programmable routing resource was detected are reconfigured for further testing in order to identify the faulty programmable routing resource. Preferably, the FPGA resources contained within the group of WUTs 52 containing the faulty resource are subdivided for inclusion in two subsequent groups of WUTs having known fault-free resources. The programmable routing resources in each subsequent group of WUTs are independently tested in the manner described above. Dependent upon the subsequent test results, the programmable routing resources within one or both of the subsequent groups of WUTs may be further reconfigured or subdivided and tested until the detected faulty programmable routing resource is identified.

Detailed Description Text (22):
Upon completion of testing of each of the FPGA resources within the initial self-testing area 16, the FPGA under test 10 is reconfigured such that the functions of the PLBs forming a portion of the working area are copied to the PLBs forming the initial self-testing area 16 and appropriately re-routed. Once completed, the copied portion of the working area becomes a subsequent self-testing area. Preferably, the initial self-testing area 16 is reconfigured as an adjacent portion of the working area, i.e., the programmed function of an adjacent portion of the working area is relocated or more specifically, copied to the initial self-testing

area 16, and the adjacent portion of the working area is reconfigured as the subsequent self-testing area. The present preferred method of roving the self-testing area 16 and reconfiguring the FPGA under test 10 is described in detail in the above noted U.S. application Ser. No. 09/405,958 incorporated herein by reference.

Detailed Description Text (23):
In accordance with the present inventive method, the FPGA resources of the subsequent self-testing area are similarly tested, detected faults within the resources identified, and their faulty modes of operation diagnosed as described above for the initial self-testing area 16. This continues until each portion of the entire FPGA under test 10, is reconfigured as a subsequent self-testing area and its resources tested and reconfigured if required. In other words, the self-testing area continuously roves around the FPGA under test 10 repeating the steps of configuring, testing, identifying, and reconfiguring so long as the FPGA 10 is in operation. Advantageously, normal operation of the FPGA 10 continues uninterrupted by the testing conducted within the self-testing areas.

Detailed Description Text (24):
In summary, the method of testing field programmable gate arrays in fault tolerant applications is carried out during normal on-line operation of the FPGA by configuring the FPGA resources into a working area and an initial self-testing area. The working area maintains normal operation of the FPGA throughout testing. Within the initial and subsequent self-testing areas, however, the FPGA resources are each tested and faulty resources identified by reconfiguring or subdividing and further testing the FPGA resources until the faulty resource is identified. Advantageously, the working area is substantially unaffected by testing, and testing time constraints are reduced since normal operation continues in the working area.

Other Reference Publication (1):
G. Gibson et al., "Boundary-Scan Access of Built in Self Test for Field Programmable Gate Arrays," Proc. IEEE Internationa ASIC Conf. pp. 57-61, Sep. 7-10, 1997.

Other Reference Publication (3):
W.K. Huang et al., "An Approach to Testing Programmable/Configurable Field Programmable Gate Arrays," Proc. IEEE VLSI Test Symp pp. 450-455, Apr. 28-May 5, 1996.

Other Reference Publication (5):
F. Lombardi et al., "Diagnosing Programmable Interconnect Systems for FPGAS," Proc ACM/SIGDA International Symp on FPGA, PP 100-106, 1996.

Other Reference Publication (6):
C. Stroud et al., "Evaluation of FPGA Resources for Built in Self Test of Programmable Logic Blocks," Proc ACM/SIGDA Inter. Symp. on FPGAS PP 107-113, 1996.

Other Reference Publication (7):
C. Stroud et al., "Built in Self Test for Programmable Logis Blocks in FPGA," Proc IEEE VLSI Test Symp, PP 387-392, Apr. 28-May 1, 1996.

Other Reference Publication (8):
C. Stroud et al., "Using ILA Testing for Bist in FPGAS," Proc IEEE International Test Conf., PP 68-75, Oct. 20-25, 1996.

Other Reference Publication (9):
C. Stroud et al., "BIST Based Diagnostics for FPGA Logic Blocks," Proc. IEEE International Test Conf., PP 539-547, Nov. 1-6, 1997.

Other Reference Publication (10):
N. Shnidman et al., "On-Line Fault Detection for Bus-Based Field Programmable Gate Arrays," IEEE Transactions on VLSI Systems, vol. 6, No. 4, Dec., 1998.

CLAIMS:

1. A method of testing resources of a field programmable gate array during normal on-line operation comprising the steps of: configuring said field programmable gate array into an initial self-testing area and a working area, said working area maintaining normal operation of the field programmable gate array; testing resources located within said initial self-testing area for faults; reconfiguring said resources located within said initial self-testing area for further testing in order to identify at least one resource having a fault detected during said initial testing step; further testing said resources located within said reconfigured initial self-testing area for faults; and repeating the steps of reconfiguring and testing until said at least one faulty resource is identified, whereby the field programmable gate array may be reconfigured to replace said at least one identified faulty resource in order to provide fault tolerant operation of the field programmable gate array.

14. The method set forth in claim 10, further comprising the step of reconfiguring the field programmable gate array to replace said at least one identified programmable logic block in order to provide fault tolerant operation only if said detected faulty mode of operation prevents a programmed operation of said at least one identified programmable logic block, whereby partially faulty programmable logic blocks may be further utilized to provide fault tolerant operation of the field programmable gate array.

15. The method set forth in claim 1, wherein the step of configuring said field programmable gate array into an initial self-testing area and a working area further includes establishing at least two testing tiles of programmable logic blocks within said initial self-testing area; and wherein said initial testing step is performed on said programmable logic blocks within said at least two testing tiles concurrently.

16. The method set forth in claim 1 further comprising the step of reconfiguring said field programmable gate array such that a portion of said working area becomes a subsequent self-testing area and at least a portion of said initial self-testing area becomes a portion of said working area.

23. An apparatus for testing the resources of at least one field programmable gate array during normal on-line operation comprising: a controller in communication with the at least one field programmable gate array for (a) configuring the at least one field programmable gate array into an initial self-testing area and a working area, said working area maintaining normal operation of the field programmable gate array, (b) testing the resources located within said initial self-testing area for faults, (c) reconfiguring the resources located within said initial self-testing area for further testing in order to identify at least one resource having a fault detected during initial testing, (d) further testing resources located within said reconfigured self-testing area, and (e) repeating the steps of (c) and (d) until said at least one faulty resource is identified.

24. The apparatus for testing the resources of at least one field programmable gate array during normal on-line operation of claim 23, further comprising at least one storage medium in communication with said controller for storing a plurality of test configurations, and usage and fault status data for each resource.

25. A field programmable gate array comprising: a plurality of programmable logic blocks; a plurality of programmable routing resources interconnecting said programmable logic blocks; said programmable logic blocks and said programmable

routing resources being initially configured as at least one initial self-testing area for testing at least a portion of said programmable logic blocks for faults, and an initial working area for maintaining normal operation of the field programmable gate array during testing; said portion of said programmable logic blocks located within said initial self-testing area being subdivided for further testing until said faulty programmable logic block is identified.

26. The field programmable gate array of claim 25, wherein said portion of programmable logic blocks of said initial self-testing area is configured as at least one initial testing tile of programmable logic blocks; and wherein said at least one initial testing tile is subdivided and its programmable logic blocks utilized in at least two subsequent testing tiles for further testing; and wherein said subsequent testing tiles are further subdivided and their programmable logic blocks utilized in subsequent testing tiles for further testing tested until said faulty programmable logic block is identified.

Previous Doc Next Doc Go to Doc#

L10: Entry 3 of 8 File: USPT Jun 3, 2003

DOCUMENT-IDENTIFIER: US 6574761 B1
TITLE: On-line testing of the programmable interconnect network in field programmable gate arrays

Abstract Text (1):
A method of self-testing the programmable routing network in a field programmable gate array (FPGA) during normal on-line operation includes configuring the FPGA into an initial self-testing area and a working area. The initial self-testing area is preferably configured to include an horizontal self-testing area primarily for testing horizontal wire segments and a vertical self-testing area primarily for testing vertical wire segments. Programmable logic blocks located within the self-testing areas are configured to function as a test pattern generator and an output response analyzer, and a portion of the programmable routing resources within the self-testing areas is configured as groups of wires under test. An exhaustive set of test patterns generated by the test pattern generator is applied to the groups of wires under test which are repeatedly reconfigured in order to completely test the programmable routing resources within the self-testing areas. The outputs of the groups of wires under test are compared by the output response analyzer and resultant fault status data for each group of wires under test is received by a controller in communication with a memory for storing the fault status data. After completely testing the programmable routing resources in one of the initial self-testing areas, the FPGA is reconfigured such that a portion of the working area becomes a subsequent self-testing area and at least a portion of one of the initial self-testing areas replaces that portion of the working area. In other words, the self-testing areas rove around the FPGA repeating the steps of testing and reconfiguring until the entire FPGA has undergone testing or continuously.

Brief Summary Text (2):
The present invention relates generally to the field of testing of integrated circuit devices and, more particularly, to a method of testing the programmable interconnect network in field programmable gate arrays.

Brief Summary Text (4):
A field programmable gate array (FPGA) is a type of integrated circuit consisting of an array of programmable logic blocks interconnected by a programmable routing network and programmable input/output cells. Programming of the logic blocks, the routing network and the input/output cells is selectively completed to make the necessary interconnections that establish one configuration thereof to provide the desired system operation/function for a particular application.

Brief Summary Text (5):
The present inventors have recently developed methods of built-in self-testing the array of programmable logic blocks and the programmable routing network in FPGAs at the device, board and system levels. These methods are set out in detail in U.S. Pat. Nos. 5,991,907; 6,003,150 and 6,202,182. The full disclosures in these patent applications are incorporated herein by reference.

Brief Summary Text (6):
In each of these prior methods, the reprogrammability of an FPGA is exploited so that the FPGA is configured exclusively with built-in self-test (BIST) logic during

off-line testing and subsequently reconfigured to its normal operating configuration. In this way, testability at every level is achieved without overhead. In other words, the BIST logic simply "disappears" when the FPGA is reconfigured for its normal system function.

Brief Summary Text (7):
In addition to these off-line testing methods, the present inventors have also recently developed methods of on-line testing and fault tolerant operation of the programmable logic blocks of FPGAs. These methods are set out in detail in U.S. Pat. 6,256,758 and pending U.S. patent application Ser. No. 09/405,958. The full disclosure of this patent and patent application are also incorporated herein by reference.

Brief Summary Text (8):
On-line testing and fault tolerant operation of FPGAs is most important in high-reliability and high-availability applications, such as, space missions or telecommunication network routers in which adaptive computing systems often rely on reconfigurable hardware to adapt system operation. In such applications, the FPGA hardware must work continuously and simply cannot be taken off-line for testing, maintenance, or repair.

Brief Summary Text (9):
When faults are detected and located in the FPGA hardware of these systems, the FPGA resources must be quickly reconfigured to continue operation in a diminished capacity or to avoid the identified faulty resources altogether. Necessarily, therefore, testing of the FPGA resources must be performed concurrently with normal system operation.

Brief Summary Text (11):
In accordance with the present invention, the method of testing field programmable gate arrays (FPGAs) is carried out during normal on-line operation of the FPGA by configuring the FPGA resources into a working area and an initial self-testing area. The working area maintains normal operation of the FPGA under test throughout testing. Within the initial and subsequent self-testing areas, however, all the resources of the programmable routing network are tested. Advantageously, the working area is substantially unaffected by the testing, and testing time constraints are reduced since normal operation continues in the working area.

Brief Summary Text (13):
In order to achieve a complete test of the programmable routing resources of the FPGA under test, the groups of wires under test include wire segments of varying lengths interconnected by configuration interconnect points. There are two basic types of configuration interconnect points, including cross-points and break-points, and each generally includes a transmission gate controlled by a configuration memory bit.

Brief Summary Text (15):
As noted above, the output patterns of the groups of wires under test are compared and test result data is generated based on the outcome of the comparison. Passing test result data is generated if the corresponding test patterns match. If a mismatch occurs, a failing test result indication or data is generated. A failing test result or mismatch may be caused by a fault in a wire segment, a configuration interconnect point or a programmable logic block of the groups of wires under test in the self-testing area.

Brief Summary Text (17):
In order to minimize the number of reconfigurations of the FPGA under test and maintain a short total testing time, parallel testing of the programmable routing resources may be utilized. Specifically, comparisons of the output patterns of the groups of wires under test may be made at several locations along the groups of

wires under test. Advantageously, one set of test patterns may be used to test several differing groups of wires per configuration. In addition, the test result data from several compared groups of wires under test may be combined utilizing an iterative comparator. Alternatively, the test result data can be routed directly to an input/output cell of the FPGA under test. Advantageously, this latter approach provides information regarding the location of the fault in the FPGA under test, as opposed to a single pass/fail test result indication for the entire test.

Brief Summary Text (18):
In accordance with an important aspect of the present invention, the self-testing area of the FPGA under test may be divided into vertical and horizontal self-testing areas. Preferably, vertical wire segments are tested utilizing the vertical self-testing area and horizontal wire segments are tested utilizing the horizontal self-testing area. To accommodate on-line testing, programmable logic blocks in both self-testing areas, vertical wire segments in the vertical self-testing area, and horizontal wire segments in the horizontal self-testing area are all designated reserved or unusable during operation of the FPGA under test. In this manner, connections between working area programmable logic blocks may be made utilizing horizontal wire segments through the vertical self-testing area and vertical wire segments through the horizontal self-testing area.

Brief Summary Text (19):
Upon completion of testing of the programmable routing resources located within the initial self-testing area, the FPGA under test is reconfigured so that a portion of the working area becomes a subsequent self-testing area, and the initial self-testing area becomes a portion of the working area. In other words, the self-testing area roves around the FPGA under test repeating the steps of reconfiguring and testing the programmable routing resources in the self-testing areas until each portion of the working area, or the entire FPGA, is reconfigured as a subsequent self-testing area and tested. As noted above, the present method of testing allows for normal operation of the FPGA under test to continue within the working area throughout testing, uninterrupted by the testing conducted within the self-testing areas.

Brief Summary Text (20):
The steps of configuring, testing, and reconfiguring the groups of wires under test and programmable logic blocks within the self-testing areas, storing the subsequent fault status data, and roving the self-testing area around the FPGA under test for further testing are necessarily controlled by a test and reconfiguration controller and an associated storage medium. In operation, the test and reconfiguration controller accesses the FPGA under test during normal system operation and configures the FPGA with one of a plurality of test configurations stored in the associated storage medium. As described above, test patterns generated within the self-testing area provide for exhaustive testing of the groups of wires under test. The test patterns are propagated along the groups of wires under test and the output patterns compared.

Drawing Description Text (3):
FIG. 1 is a schematic block diagram of an apparatus for testing the programmable interconnect network of a field programmable gate array;

Drawing Description Text (4):
FIG. 2 is an illustration of the FPGA under test configured into an initial self-testing area and a working area wherein the working area maintains normal operation of the FPGA under test;

Drawing Description Text (7):
FIG. 5A is a schematical illustration showing a typical configuration interconnect point of a field programmable gate array;

Drawing Description Text (8):
FIG. 5B is a schematical illustration showing a typical cross-point configuration interconnect point of a field programmable gate array;

Drawing Description Text (9):
FIG. 5C is a schematical illustration of a typical break-point configuration interconnect point of a field programmable gate array;

Drawing Description Text (10):
FIG. 6 is a schematical illustration of a typical programmable logic block of a field programmable gate array;

Drawing Description Text (15):
FIG. 10 is an illustration of the preferred FPGA under test with the initial self-testing area subdivided into a vertical self-testing area and an horizontal self-testing area; and

Drawing Description Text (16):
FIG. 11 is a wiring schematic illustrating the preferred method of testing cross-point CIPs utilizing FPGA resources within both a vertical self-testing area and an horizontal self-testing area.

Detailed Description Text (3):
A typical field programmable gate array (FPGA) generally consists of an array of programmable logic blocks interconnected by a programmable routing network and programmable input/output cells or boundary-scan ports (most FPGA's feature a boundary scan mechanism). Such structures are, for example, featured in the Lucent ORCA programmable function units, in the Xilinx XC4000 configurable logic block, and in the ALTERA FLEX 8000 logic element. In accordance with the method of the present invention, the resources of the programmable routing network of the FPGA under test are completely tested during normal operation by configuring the FPGA into a working area and a self-testing area. Advantageously, the working area is substantially unaffected by the testing conducted within the self-testing area.

Detailed Description Text (4):
As shown in schematic block diagram in FIG. 1, the steps of configuring, testing, roving, and reconfiguring the resources of an FPGA under test 10 are necessarily controlled by a test and reconfiguration controller 12. In the preferred embodiment, an external test and reconfiguration controller 12 is utilized because present commercially available FPGAs do not allow internal access to their configuration memory. Accordingly, a configuration decompiler tool of a type known in the art is utilized to determine the intended function or mode of operation of the FPGA resources. Alternatively, this information may be extracted from the design stage and made available to the controller 12. It should be appreciated by those skilled in the art that any controller, e.g., internal or external to the FPGA, could be utilized with an FPGA that allows for internal access to its configuration memory and that a single test and reconfiguration controller is capable of controlling several FPGAs. For purposes of illustration of the present preferred embodiment of the invention, however, a one-to-one controller to FPGA ratio is utilized.

Detailed Description Text (5):
The preferred controller 12 may be implemented on an embedded microprocessor in communication with a storage medium or memory 14 for storing the configurations and test data. In operation, the controller 12 accesses the system platform including FPGA under test 10 through its boundary-scan ports in a known manner such that access is transparent to the normal function of the FPGA 10. Advantageously, this approach allows for complete on-line testing during normal operation of the FPGA 10 under test. The controller 12 and memory 14 further exchange and store fault status data for the programmable routing resources.

Detailed Description Text (6):
In accordance with the present inventive method, the FPGA under test 10 is initially configured by the controller 12 into an initial self-testing area 16 and a working area 18 as shown in FIG. 2. The working area 18 maintains normal operation of the FPGA under test 10 throughout testing. Within the initial self-testing area 16, the resources of the programmable routing network are exhaustively tested. The programmable routing resources include both global routing resources for carrying signals amongst the array of programmable logic blocks (PLBs), and local routing resources for carrying signals into and out of the PLBs. For example, the typical global and local routing resources associated with a single PLB are shown in FIG. 3 and are discussed in more detail below.

Detailed Description Text (7):
As shown in FIG. 4, a first group of PLBs within the initial self-testing area 16 are configured to include a test pattern generator (TPG) 20 and an output response analyzer (ORA) 22, and a portion of the programmable routing resources are configured to include at least two groups of wires under test (WUTs) 24, 26. The first group of WUTs 24 may include wire segments 27, 28, 29, 30, 31 and 32, configurable or configuration interconnect points (CIPs) 33, 34 and 35, and PLBs 36, 37. Similarly, a second group of WUTs 26 may include wire segments 38, 39, 40, 41 and 42 and CIPs 43, 44 and 45.

Detailed Description Text (8):
During testing, exhaustive test patterns generated using the TPG 20 are propagated along the groups of wires under test 24, 26. The outputs of the groups of WUTs 24, 26 are compared by the ORA 22 to determine whether a fault exists within either group of WUTs 24, 26. A match/mismatch result of the comparison performed by the ORA 22 is communicated as a pass/fail test result or fault status data through the boundary-scan ports of the FPGA (not shown) to the controller 12 for storage in memory 14. The operation of the TPG 20 and ORA 22 in testing the groups of WUTs 24, 26 is similar to the built-in self-test techniques utilized and described in detail in the above-noted pending U.S. patent applications.

Detailed Description Text (19):
Alternately, the various ORA outputs can be routed directly to the boundary-scan ports to retrieve the results as indicated above. This provides more information regarding the location of the fault in the FPGA as opposed to a single pass/fail test indication for the entire test resulting from the use of an iterative comparator.

Detailed Description Text (20):
As indicated above, FIG. 3 illustrates a simplified view of the routing busses associated with a single PLB designated numeral 76 in an ORCA 2C series FPGA. Horizontal and vertical busses are denoted by h and v, respectively. The suffixes x1, x4, xH, and xL indicate wire segments that extend across 1 PLB, 4 PLBs, half the PLB array, and the full length of the PLB array, respectively, before encountering a break-point CIP or a boundary-scan point of the FPGA (not shown). Direct busses provide connections between adjacent PLBs. The four direct busses are designated dn, ds, de, and dw denoting direct north, south, east, and west, respectively. For every PLB there are two sets of vertical x1 busses and two sets of horizontal x1 busses, designated vx1w, vx1e, hx1n, and hx1s. Several CIPs are available to establish different connections among the wire segments as shown by circle and diamond-shaped symbols, The diamond-shaped symbol 77 of a break-point CIP on a 4-bit bus represents a group of 4 individual break-point CIPs. Similarly, and as shown in FIG. 9A, a circle-shaped symbol 78 denoting a cross-point CIP at the intersection of a vertical 4-bit bus with an horizontal 4-bit bus represents a group of 4 individual cross-point CIPs between corresponding wires in the two busses. The square-shaped symbol 79, on the other hand, at the intersection of a 5-bit direct bus with a 4-bit x1 bus represents a more flexible matrix of cross-point

CIPs shown in FIG. 9B.

Detailed Description Text (21):
The preferred fault model utilized to test the resources of the programmable interconnect network of a typical FPGA, includes CIPs stuck-closed (stuck-on) and stuck-open (stuck-off), wire segments stuck at 0 or 1, open wire segments, and shorted wire segments. Detecting the CIP faults also detects stuck-at faults in the configuration memory bits that control the CIPs as shown generally in FIG. 5A. For generality, both wired-AND and wired-OR faults are considered as possible behavior for shorted wire segments. A stuck-closed CIP creates a short between its two wires.

Detailed Description Text (22):
Since detailed layout information regarding the adjacency relationships between wire segments is typically not available, only rough physical data available in FPGA data books, for example, is utilized to determine bunches of wire segments for testing. A bunch of wire segments are wire segments that may have pair-wise shorts; but not every wire segment is necessarily adjacent to every other wire segment in the bunch. For example, all the vertical wire segments located between two adjacent PLB columns may be treated as a bunch even if not all shorts are physically feasible. Advantageously, this makes the preferred testing method layout-independent and allows the bus rotations, which make the adjacency relations among wire segments of the same bunch change, to be ignored during testing.

Detailed Description Text (25):
In accordance with the present preferred method shown in FIG. 10, the initial self-testing area 16 of the FPGA under test 10 may be divided into a vertical self-testing area 80 and an horizontal self-testing area 81. The vertical self-testing area 80 is primarily utilized to test vertical routing resources or wire segments, and the horizontal self-testing area 81 is primarily utilized to test horizontal routing resources or wire segments. To accommodate operation of the FPGA under test 10 during testing, spare programmable logic blocks in both self-testing areas, horizontal wire segments in the horizontal self-testing area 80, and vertical wire segments in the vertical self-testing area 81 are all designated reserved or unusable. In accordance with an important aspect of the present invention, connections between divided working area PLBs are made utilizing horizontal wire segments through the vertical self-testing area 80 and vertical wire segments through the horizontal self-testing area 81.

Detailed Description Text (27):
Testing each of the cross-point CIPs of the FPGA under test 10 in this manner would require a different test configuration for every possible pair of positions of the vertical and horizontal self-testing areas 80 and 81. In accordance with the present preferred method, therefore, only a subset of the cross-point CIPs are tested on every full horizontal sweep of the self-testing area. More specifically, the horizontal self-testing area 81 is maintained in a fixed position and only the subset of cross-point CIPs in the current intersection of the vertical self-testing area 80 and the horizontal self-testing area 81 are tested. For every full sweep of the vertical self-testing area 80, however, the position of the horizontal self-testing area 81 is different. Advantageously, this preferred method substantially limits the increase in fault latency and overall testing time.

Detailed Description Text (28):
Upon the completion of testing the programmable routing resources located within the initial self-testing area 16, the FPGA under test 10 is reconfigured such that the functions of the PLBs forming a portion of the working area 18 are copied to the PLBs forming the initial self-testing area 16. Once completed, the copied portion of the working area becomes a subsequent self-testing area. Preferably, the initial self-testing area 16 is reconfigured as an adjacent portion of the working area 18, i.e., the programmed function of an adjacent portion of the working area

18 is relocated or more specifically, copied to the initial self-testing area 16, and the adjacent portion of the working area is reconfigured as the subsequent self-testing area.

Detailed Description Text (29):
In accordance with the preferred present inventive method described above, the subsequent self-testing area may similarly be divided into vertical and horizontal self-testing areas 81 and 82 (as shown in FIG. 10) if desired. Further, the step of testing the programmable routing resources within the subsequent testing area is then repeated. This continues until each portion of the working area 18, or the entire FPGA under test 10, is reconfigured as a subsequent self-testing area and its programmable routing resources tested. In other words, the self-testing area roves around the FPGA under test 10 repeating the steps of testing and reconfiguring the programmable routing network until the entire FPGA has undergone testing. Advantageously, normal operation of the FPGA under test 10 continues uninterrupted by the testing conducted within the self-testing areas.

Detailed Description Text (30):
The present preferred method of roving the self-testing area 16 or reconfiguring the FPGA under test 10 is described in detail in the above-mentioned U.S. patent application Ser. No. 09/405,958, incorporated herein by reference.

Detailed Description Text (31):
In summary, the method of testing field programmable gate arrays (FPGAs) is carried out during normal on-line operation of the FPGA by configuring the FPGA resources into a working area and an initial self-testing area. The working area maintains normal operation of the FPGA under test throughout testing. Within the initial and subsequent self-testing areas, however, all the programmable routing resources are exhaustively tested. Advantageously, the working area is substantially unaffected by the testing and testing time constraints are reduced since normal operation continues in the working area.

Other Reference Publication (3):
W.K. Huang et al., "An Approach to Testing Programmable/Configurable Field Programmable Gate Arrays," Proc. IEEE VLSI Test Symp pp. 450-455, Apr. 28-May 5, 1996.

Other Reference Publication (5):
F. Lombardi et al., "Diagnosing Programmable Interconnect Systems for FPGAS," Proc. ACM/SIGDA International Symp on FPGA, pp 100-106, 1996.

Other Reference Publication (6):
C. Stroud et al., "Evaluation of FPGA Resources for Built In Self Test of Programmable Logic Blocks," Proc ACM/SIGDA Inter. Symp. on FPGAS pp 107-113, 1996.

Other Reference Publication (7):
C. Stroud et al., "Built In Self Test for Programmable Logis Blocks in FPGA," Proc IEEE VLSI Test Symp, pp 387-392, Apr. 28-May 1, 1996.

Other Reference Publication (8):
C. Stroud et al., "Using ILA Testing for BIST in FPGAS," Proc IEEE International Test Conf., pp 68-75, Oct. 20-25, 1996.

Other Reference Publication (9):
C. Stroud et al., "BIST Based Diagnostics for FPGA Logic Blocks," Proc. IEEE International Test Conf., pp 539-547, Nov. 1-6, 1997.

Other Reference Publication (10):
N. Shnidman et al., "On-Line Fault Detection for Bus-Based Field Programmable Gate Arrays," IEEE Transactions on VLSI Systems, vol. 6, No. 4, Dec., 1998.

CLAIMS:

1. A method of testing programmable routing resources of a field programmable gate array during normal on-line operation comprising the steps of: configuring said field programmable gate array into an initial self-testing area and a working area, said working area maintaining normal operation of the field programmable gate array; testing said programmable routing resources located within said initial self-testing area; and roving said initial self-testing area by reconfiguring said field programmable gate array such that a portion of said working area becomes a subsequent self-testing area and at least a portion of said initial self-testing area becomes a portion of said working area.

3. The method set forth in claim 1, wherein the step of configuring said field programmable gate array into an initial self-testing area and a working area further includes configuring a first group of programmable logic blocks within said initial self-testing area to function as a test pattern generator and an output response analyzer, and a portion of said programmable routing resources within said initial self-testing area as at least two groups of wires under test.

7. The method set forth in claim 3, wherein the step of configuring a portion of said programmable routing resources within said initial self-testing area further includes utilizing a group of wire segments and a group of configuration interconnect points of said programmable routing resources to form said at least two groups of wires under test.

8. The method set forth in claim 7, wherein the step of configuring said field programmable gate array into an initial self-testing area and a working area further includes configuring a second group of programmable logic blocks within said initial self-testing area to pass said test patterns there through, said second group of programmable logic blocks forming a portion of said at least two groups of wires under test, whereby local routing resources of said programmable routing resources are tested.

11. The method set forth in claim 9 further comprising the step of repeatedly reconfiguring said programmable routing resources within said initial self-testing area so each wire segment and configuration interconnect point becomes one of said at least two groups of wires under test at least once during testing.

14. The method set forth in claim 12, wherein the step of roving said initial self-testing area includes reconfiguring said field programmable gate array such that a portion of said working area becomes a subsequent horizontal or vertical self-testing area and said initial horizontal or vertical self-testing area becomes a portion of said working area.

15. A method of testing a field programmable gate array including programmable routing resources and programmable logic blocks during normal on-line operation comprising the steps of: configuring said field programmable gate array into an initial self-testing area and a working area maintaining normal operation of said field programmable gate array; applying test patterns generated by said programmable logic blocks to said programmable routing resources configured as groups of wires under test within said initial self-testing area; utilizing said programmable logic blocks to compare outputs of said groups of wires under test within said initial self-testing area; and reconfiguring said field programmable gate array such that a portion of said working area becomes a subsequent self-testing area and at least a portion of said initial self-testing area becomes a portion of said working area.

17. The method set forth in claim 15 further comprising the step of repeatedly reconfiguring said programmable routing resources so each wire segment and

configuration interconnect point within said initial self-testing area is utilized in one of said groups of wires under test at least once during testing.

18. A method of testing a field programmable gate array including programmable routing resources and programmable logic blocks during normal on-line operation comprising the steps of: configuring said field programmable gate array into an initial horizontal self-testing area, an initial vertical self-testing area, and a working area maintaining normal operation of said field programmable gate array; applying test patterns generated by said programmable logic blocks to a portion of said programmable routing resources configured as groups of wires under test within said initial self-testing areas; utilizing said programmable logic blocks to compare outputs of said groups of wires under test within said initial self-testing areas; and reconfiguring said field programmable gate array such that a portion of said working area is utilized in forming a subsequent self-testing area and at least a portion of one of said initial self-testing areas becomes a portion of said working area.

22. The method set forth in claim 18, wherein the step of applying test patterns to a portion of said programmable routing resources configured as groups of wires under test further includes utilizing a group of wire segments and a group of configuration interconnect points of said programmable routing resources to form said groups of wires under test.

23. An apparatus for testing programmable routing resources of a field programmable gate array during normal on-line operation comprising: a controller in communication with said field programmable gate array for (a) configuring said field programmable gate array into an initial self-testing area and a working area, said working area maintaining normal operation of the field programmable gate array, (b) for testing said programmable routing resources located within said initial self-testing area, and (c) roving said initial self-testing area by reconfiguring said field programmable gate array such that a portion of said working area becomes a subsequent self-testing area and at least a portion of said initial self-testing area becomes a portion of said working area; and a storage medium in communication with said controller for storing a plurality of test configurations and fault status data.

24. A field programmable gate array comprising: a plurality of programmable logic blocks; a plurality of programmable routing resources interconnecting said programmable logic blocks; a plurality of input/output cells; said programmable logic blocks and said programmable routing resources being initially configured as an initial self-testing area for testing a portion of the programmable routing resources within said initial self-testing area, and an initial working area for maintaining normal on-line operation of the field programmable gate array during testing; and said programmable logic blocks and said programmable routing resources being subsequently configured as a subsequent self-testing area for testing a different portion of the programmable routing resources within said subsequent self-testing area, and a subsequent working area for maintaining normal on-line operation of the field programmable gate array during subsequent testing.

25. The field programmable gate array of claim 24, wherein a portion of said programmable logic blocks within said initial self-testing area are configured to function as a test pattern generator and an output response analyzer, and a portion of said programmable routing resources within said initial self-testing area are configured as at least two groups of wires under test.

26. The field programmable gate array of claim 25, wherein said test pattern generator generates an exhaustive set of test patterns for testing said at least two groups of wires under test; and said output response analyzer compares outputs of said at least two groups of wires under test and produces fault status data for said at least two groups of wires under test.